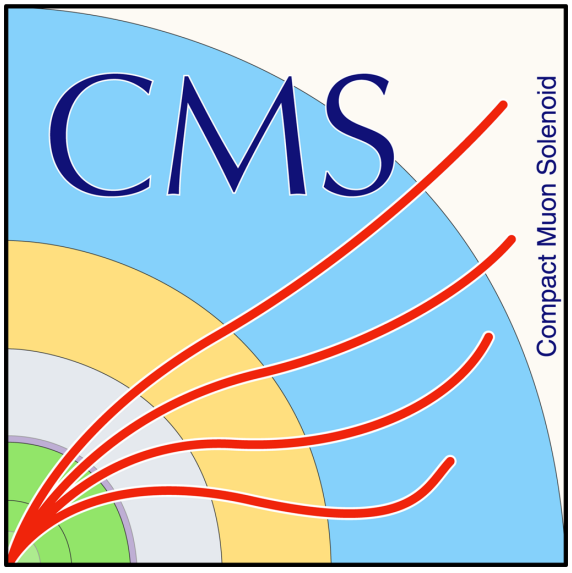
Installing Rocks 7 onto a High Throughput Computing Cluster

April 12, 2019



Open Science Grid

Contents

1 Background

Florida Tech's High Energy Physics Lab's High Throughput Computing Cluster is a sophisticated piece of computing machinery that has the capacity to not only aide research at Florida Tech, but around the world. In addition to serving as the HEP group's primary storage center, the cluster is associated with the Open Science Grid, which allows researchers from across the globe to run compute jobs on our machine. The cluster is glued together with the cluster-building Linux distribution Rocks. Rocks has its base in CentOS with additional features that ease the creation of computing clusters.

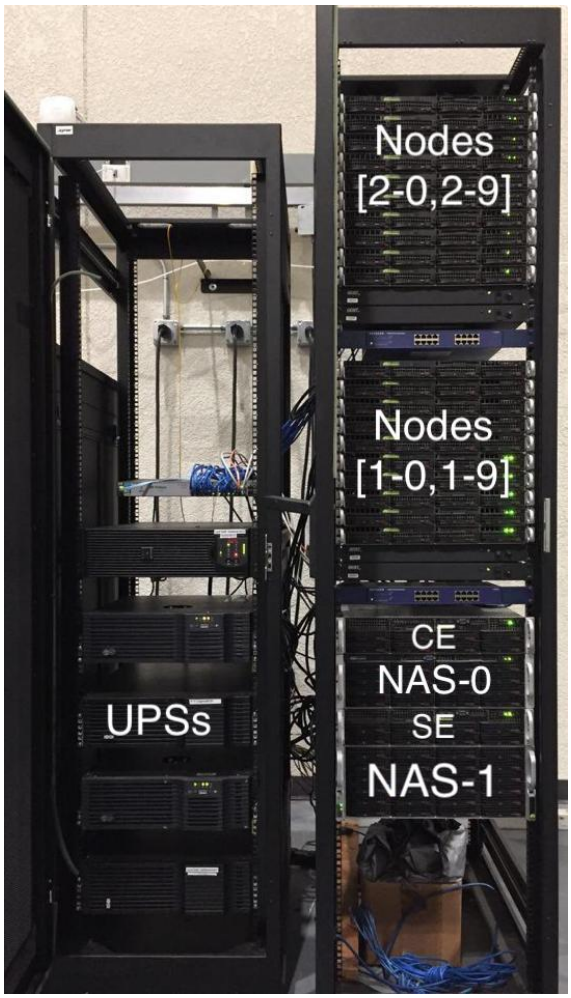


Figure 1: A labeled picture of the cluster.

Table 1: Descriptions of each labeled component of the cluster.

Compute Element (CE)	The CE is the head node of the cluster. It manages all the other components, and is where administrators spend most of their time.
Storage Element (SE)	The SE is responsible for managing data transfers from CERN. Current scientific data is needed by researchers hoping to run jobs on the cluster.
NAS-0	NAS-0 is the smaller of the cluster's two storage units. Its 16 750GB hard drives store the home directories of all the users.
NAS-1	NAS-1 is the larger of the cluster's two storage units. The HEP group researchers use its 50TB capacity to store and backup their scientific data.
Nodes	The cluster's 20 compute nodes each have 8 processors ready to compute incoming jobs.
Uninterruptible Power Supplies (UPSs)	The UPSs are large battery banks that provide emergency power to the cluster in the event of brief power flickers.

2 Preparation

Before a major overhaul of any computer system, it is imperative that it first be properly prepared. The key components must be backed up so that they may be available for reference while setting up the new system, and the continued operation of tangential components not participating in the update must be ensured.

2.1 Backup Data

One of the most important preliminary steps is to first backup all the data on the devices that are being updated. Onto NAS-1, we backed up the contents of the CE, omitting `/mnt` and `/cvmfs`, the OS (250 GB) and data (1 TB) drives of the SE, and both the server and data partitions of NAS-0. For the CE and SE, `/mnt` was excluded because we only want to back up data on the CE and SE rather than the data of other systems that may be mounted. `/cvmfs` was excluded for the CE because it is externally mounted from CERN, thus eliminating the need for a local backup. Unlike the CE and SE, NAS-0 does not have NAS-1 mounted, so NAS-0's files had to be copied remotely via the CE, on which NAS-1 is mounted, before being stored in NAS-1. Since there is already an older backup of NAS-0 on NAS-1 and there is not enough space on NAS-1 to fully copy another backup of NAS-0, all of the newer files on NAS-0 were manually copied into the old backup of NAS-0, effectively updating it.

Backup the CE (executed in CE)

```
rsync -av --exclude '/mnt*' --exclude '/cvmfs' / /mnt/nas1/CEBackup-20180803
```

Backup the SE (executed in SE)

```
rsync -av --exclude '/mnt' / /mnt/nas1/SEBackup-20180803
```

Backup the server partition of NAS-0 (executed in CE)

```
rsync -av --exclude '/nas0' root@10.1.255.234:/mnt/nas1/nas0-bak-20180730/NAS-0
```

Ideally, the entirety of one compute node would also be backed up at this point, but due to catastrophic hardware failures preventing the nodes from being run (one of the big Tripplite UPSs AND the APC UPS have failed, leaving me with only one UPS), I'm going to come back to them later.

2.2 Burn Boot Disc

Once all of the data is safely stored away, a bootable DVD must be created. The ISO for the Rocks kernel can be found on their website. Although there are several rolls (software package ISOs) labeled **required** on the website, only the "kernel" roll is required; the other rolls will be downloaded and installed from the installer on the kernel roll. Because the firmware of the CE does not support USB boot, the 1GB ISO must be burned onto a DVD.

2.3 Ensure NAS-1 Remains Operational

Because NAS-1 does not need to be updated, and since everyone needs to access their data, it must be kept running and accessible during the rebuild process. Since NAS-1 has never technically been part of the cluster with Rocks, it is independently addressable via its own IP address. This independent addressability permits NAS-1 to be mounted and remotely accessed via ssh without the rest of the cluster. It just needs to be left on and connected to the router.

3 Installation

3.1 Loading the Installer on the CE

The kernel DVD made earlier will be used to boot the CE into the Anaconda installer. Before the CE can be booted into the DVD, its boot order must configured to prioritize booting from its disc drive. It's BIOS can be accessed by spamming the DELETE key on startup.

Once the CE's boot order has been properly configured, the kernel DVD may be inserted and the CE booted. The CE will then boot into the Anaconda installer after performing several checks.

3.2 Public Network Configuration

The first step of the installation process is configuring the network so that the other necessary software packages can be installed. Select *Network & Host Name*, and ensure that the 10GB/s ethernet connection (ethernet device usually beginning with `enp10`) is turned on. Further configuration takes place behind the "Configure..." button:

- Under the "IPv4 Settings" tab:
  - "Method" is set to **Manual**.
  - The IP is set to the CE's IP: **163.118.42.1**.
  - The netmask is set to **25** to encompass the IP range 163.118.42.1-163.118.42.127.
  - The gateway IP is **163.118.42.126**, and it was found running `ip route` in NAS-1.
  - A DNS server, such as **8.8.8.8**, must be provided so that hostnames, including the CE's, can be resolved.
- Under the "IPv6 Settings" tab:
  - "Method" is set to **Link-Local Only**.

With this configuration in place, select *Save* and switch on the ethernet connection. If the "Current host name" text box contains "uscms1.fltech-grid3.fit.edu", the hostname of the cluster, then the CE should be connected to the internet, and the installation can continue!

3.3 Private Network Configuration

With internet access obtained, all the previously grayed-out options are now full of color! The next step is to configure the *Cluster Private Network*. The only thing that must be done here, is to verify that the correct ethernet port is selected. The CE has a 10GB/s ethernet connection for internet use, it has already been configured, and a 1GB/s ethernet connection (ethernet device usually beginning with `enp6`) for internal communication with the other cluster components. The 1GB/s connection should be selected by default, and the other default settings are correct. This section can usually be closed without having altered any of the settings.

3.4 Rolls

Now the rolls can be installed! Rolls are software packages that are installed alongside the operating system. Select "ROCKS ROLLS". Here, rolls may be installed remotely from the Rocks website. Select *List Available Rolls* to generate the roll list. Select the following rolls:

- base
- CentOS
- core
- ganglia
- htcondor
- kernel
- perl
- python
- Updates-CentOS-<version>

Once all the desired rolls are selected, select *Add Selected Rolls*. After the rolls appear in the "Selected Rolls" section, select *Done*.

3.5 Cluster Configuration

Select *CLUSTER CONFIG* to configure the cluster. The fields ought to resemble those in Table 2.

Table 2: A recommendation on how to fill in the cluster configuration fields.

Cluster Name	USCMS-FIT-Grid
Contact	<administrator email>
Project URL	`https://research.fit.edu/hep/hohlmann-research-group/grid-cluster`
Latitude/Longitude	N28.0622 W80.6237
Certificate Organization	Florida Institute of Technology
Certificate Locality	Melbourne
Certificate State	Florida
Certificate Country	US
NTP Servers	pool.ntp.org

After the fields have been verified to be correct, select *Done*.

3.6 Partitioning

WARNING

When installing a new operating system, be SURE that the destination drive is correct. Hot spares and data partitions will show up alongside OS drives in the installer.

Now the CE must be partitioned. Select *INSTALLATION DESTINATION*. Select the larger of the two drives (the smaller one is the hot spare and it is NOT TO BE TOUCHED), and select *I will configure partitioning.* under "Other Storage Options". Select *Done* to be kicked over to Anaconda's partitioning system.

First, the old partition, the CentOS 6 one, was selected, then removed by selecting the "-" at the bottom of the window and checking the box for deleting all file systems. Then the *Click here to create them automatically* option was selected.

Since NAS-0 holds the home directories, the default `/home` partition is renamed to `/export`, and a new `/var` partition is created. The partition scheme ought to match what is shown in Table 3.

Table 3: The partitioning configuration for a fresh CE.

Partition Name	Allocated Memory
`/export`	100 GiB
`/var`	100 GiB
`/boot`	default
`swap`	default
`/`	remaining

Once everything is together, select *Done* and carefully review the changes. If the changes are correct, select *Accept Changes*.

3.7 Commit

WARNING

There is NO "Are you sure?" when *Begin Installation* is pressed!

If everything is undoubtedly confirmed (to the best of your ability), and you're absolutely ready to destroy everything and build it back up, say a quick prayer to your deity of choice and select *Begin Installation*.

Create a root password when prompted, and do NOT setup a user.

After the installation is complete, reboot as per the instructions. After reboot do not enter any further configuration; simply continue. When presented with the login prompt, login as root. Congratulations! Rocks has been installed onto the CE!

4 Unification

Now that Rocks is installed onto the CE, it can be installed onto the other cluster components so that they may be brought into the fold.

4.1 Nodes

To begin, login to the CE as root and run `insert-ethers`, the program that routes network traffic from the nodes to the Rocks database for processing. From the menu, select *Compute*. The "Inserted Applicances" screen that appears should initially be blank; nodes must be turned on for them to be recognized and configured.

Insert the Rocks kernel disc into the first node, `compute-1-0`, and restart it. `compute-1-0` should appear in the "Inserted Applicances" menu. The `()` in the rightmost column will be blank for now. A star (`*`) will appear when the node requests a kickstart file from the CE. The node will boot into the Anaconda installation for Rocks 7.

4.1.1 Installing Rocks onto a Compute Node

The CE hosts a kickstart file that the nodes use to configure their installation. In order to fetch this file on startup, the nodes must be configured for PXE network booting. This is achieved by ensuring PXE is at the top of the boot order in the BIOS.

NOTE

If the CE is on while this is being completed, when the nodes restart, they will immediately grab the kickstart file from the CE regardless if `insert-ethers` is running or not. If this is undesired behavior, shutdown the CE before configuring the nodes.

To enter a node's BIOS, spam the DELETE key on startup until "Entering SETUP" is displayed. To modify the boot order, navigate to the "Boot" tab and follow the on screen directions to adjust the boot order so that it matches what is shown in Table 4.

Table 4: The boot order of the nodes.

1	`PCI BEV: IBA GE Slot 0400 v1236`	ethernet (PXE boot)
2	`IDE CD`	boot from disc drive
3	`PCI SCSI`	internal hard drive

4.1.2 Incorporate NAS-0

Now that the nodes are in, let's see about bringing NAS-0 into the fold. Similar to the nodes, its boot order must also be changed so that ethernet boot is enabled. After spamming the DELETE key on startup and navigating to the boot menu of the BIOS, the boot order was arranged as is shown in Table 5.

Table 5: The proper boot order of NAS-0.

1	`PCI BEV: IBA GE Slot 0600 v1236`	ethernet (PXE boot)
2	`IDE CD`	boot from disc drive
3	`PCI SCSI: 3ware Storage Controller`	boot from RAID card

After that configuration, `insert-ethers` was run on the CE and *NAS Appliance* was selected. Now we're ready to save our boot options on NAS-0 and reboot! Let's see if it works.

We're in! NAS-0 has booted into Rock's Anaconda installer via the kickstart file. `insert-ethers` also recognizes NAS-0! Let's see about going through the installer.

The most pressing issue is *Installation Destination*. This section is a bit tricky because all the disks attached to the RAID card are also displayed and care must be taken to not install the OS on a RAID disk. There are 13 disks listed in the "Local Standard Disks" section (`sd[a-m]`) and one disk listed in the "Specialized & Network Disk" (`mpatha`). That's a total of 14 disks out of NAS-0's 16. I'm guessing the two unlisted disks are the two I dedicated to housing the OS, which is unfortunate because those are the only two we need. Hmm.

Lol, the OS mirror is degraded; one of the drives failed. That's fine; we're just booting into NAS-0 normally to replace the drive, then we should be good to go. We'll also be making note of what all the various groups are named to we can keep track of them in the installer.

REPLACING NAS-0 OS DRIVE: NAS-0's RAID card is accessible via `tw_cli`. We investigated the health of the OS RAID by running `tw_cli /c0/u0 show`, which reported that the array was degraded; unit `u0-0` was DEGRADED, while unit `u0-1`, said to be in port 1, was OK. To formally remove the degraded drive, we ran `tw_cli maint remove c0 p0`, which said that the port was empty. This makes sense, since unit `u0-0` does not have a port listed. We are going to simply remove the drive 0:0 and throw a new drive in its place.

4.1.3 Incorporate SE

The SE acts as a node (before, it was named `compute-0-0`). So we'll see about adding it in as a node. First, like the nodes, the SE's boot order must be modified. The procedure is the same as the nodes, shown in Table 4. However, on the SE, there are two ethernet ports listed, and we chose the one titled `PCI BEV: IBA GE Slot 0600 v1324`.

Now that the boot order has been fixed, we can run `insert-ethers` on the CE and select *Compute*. Go back to the SE, and select "Save and Exit" to restart with the new boot order. The SE will request the kickstart file and load the Anaconda installer.

ASIDE

The installer needs to be told where to install Rocks 7. There are two disks on display, a 250GB `sda` and a 1TB `sdb`. I'm gonna boot into the current disk to investigate how these drives are put together so I can make an informed decision regarding how to install ROCKS 7 (I'm learning from my previous mistakes born of haste). I changed the boot order to prioritize the hard drive to get where I need to go. The little 250GB drive is, in fact, the OS drive, and the large 1TB drive is a dedicated backup drive, so we will be installing ROCKS 7 on just the 250GB drive. We changed the boot order back to favor PXE and restarted.

The installer decided to automatically begin the installation. Some `/dev/sda` stuff flashed by without any `/dev/sdb` stuff, so it appears to have figured itself out.

NOTE

Do not forget to set the root password during the install! If it is not entered before the install is completed, it can be set using Rocks commands on the CE:

- `rocks set host sec_attr compute-x-x attr=root_pw`
- `rocks sync host sec_attr compute-x-x`

4.1.4 Configuring the Nodes

NOTE

The root password of the nodes may not have been set (they cannot be directly accessed from themselves). They can be accessed, however, from the CE. To set the root password, simply ssh into a node from the CE with `ssh compute-[0-1]-[0-9]` and run `passwd` to set the root password.

Since the UPSs are having problems providing power for all the nodes at once, the nodes will be configured in two equally sized groups: `compute-*-[0-4]` and `compute-*-[5-9]`.

5 Configuration

5.1 Enable SSH

`ssh` communication with the cluster is paramount in its efficient administration. Upon initial installation, the `sshd` service ought to already be running. This can be verified with `service sshd status`. If it's not running, it can easily set up after a quick internet search.

5.2 Automatic Mounting of Key Components

5.3 HTCondor

HTCondor is the cluster's job submission and routing software; it receives jobs and sends them to the nodes to be computed. Since the HTCondor roll was installed alongside Rocks, HTCondor is available on the CE straight away; `condor_status` shows that HTCondor recognized the nodes' CPUs. Unfortunately, however, job submission does not immediately work. Submitted jobs are held without begin computed.

5.4 Setup SE

Several different pieces of software must be installed and configured on the SE.

Figure 2: A list of all the software that must be installed on the SE.

HDFS	Hadoop Distributed File System for the SE's data drive
xrootd	data storage system
PhEDEx	CERN data routing
squid	caching proxy to aide data transfers

6 Troubleshooting Log

6.1 Boot Loader Install Failed

During the first attempted installation of Rocks 7 onto the CE, the boot loader failed to install. Selecting *no* on the dialogue box will present an "unknown error" screen with a log and a debug option. Selecting "debug" kicks the screen over to the terminal screen.

Navigating to the "program-log" tab, the following errors may be found after the installer attempted to run the following command:

```
    grub2-install --no-floppy /dev/sda
```

```
grub2-install:  warning:  Attempting to install GRUB to a disk with multiple
partition labels.  This is not supported yet...
grub2-install:  error:  embedding is not possible, but this is required for
cross-disk install.
```

The decision to try to install to `/dev/sda` is strange because according to `fdisk -l /dev/sda` and `fdisk -l /dev/sdb`, both run in the "shell" tab, reveal that `/dev/sdb1` is marked for boot, whereas no such marking is present in the single part of `/dev/sda`.

Attempting to install GRUB onto `/dev/sda` or `/dev/sdb` yields the same result:

```
   grub2-install --no-floppy /dev/sda

   grub2-install  --no-floppy /dev/sdb
```

```
grub2-install:  error:  /usr/lib/grub/i386-pc/modinfo.sh doesn't exist.  Please
specify --target or --directory.
```

Investigating that path, `/usr/lib/grub` comes up empty. To find where that file is actually located, since it has to be somewhere if the installer is running that command, the following command was run:

```
    find / -name modinfo.sh
```

```
/mnt/sysimage/boot/grub2/i386-pc/modinfo.sh

/mnt/sysimage/usr/lib/grub/i386-pc/modinfo.sh
```

I'm trying to run the installer's command again while specifying the location of the file it needs:

```
    grub2-install --no-floppy --directory='/mnt/sysimage/boot/grub2/i386-pc'
    /dev/sda
```

```
grub2-install:  error:  cannot open '/mnt/sysimage/boot/grub2/i386-pc/kernel.img':
No such file or directory.
```

Hmm, let's see if that file exists anywhere with:

```
    find / -name kernel.img
```

```
/mnt/sysimage/usr/lib/grub/i386-pc/kernel.img
```

Ah ha! Let's run the installer's command again, but the directory will be changed to `/mnt/sysimage/usr/lib/gr`

```
grub2-install --no-floppy --directory='/mnt/sysimage/usr/lib/grub/i386-pc'
/dev/sda
```

```
/mnt/sysimage/usr/lib/grub/i386-pc/kernel.img
```

Now I'm getting the same error as the installer! Let's try with `/dev/sdb`:

```
grub2-install --no-floppy --directory='/mnt/sysimage/usr/lib/grub/i386-pc
/dev/sdb'
```

```
Installation finished.  No error reported.
```

Exciting news!

Since the installer crashed, the CE needs to be rebooted. Be sure to swiftly eject the disc when it first begins to power on so that it tries to boot normally. It will boot to grub rescue mode because, while grub is installed, it is not configured correctly.

Running the `ls` command in grub rescue will show that there are two drives, (hd0) and (hd1), and `set` will reveal that it is trying to boot into `(hd0)`. Running `ls (hd0)` will show that its file system is unknown, while `ls (hd1)` shows a recognizable one, such as ext2. The file system of (hd0) is unknown because it is the hot spare for the RAID; grub is trying to boot into the hot spare, just as it was trying to do during installation.

The `set` command said that the variable `prefix` was set to `(hd0)/boot/grub2` and `root` was set to `hd0`. To point those at the right place I ran the following two commands:

```
set prefix=(hd1)/boot/grub2

set root=hd1
```

Instructions online say after fixing the variables to try loading the normal module with `insmod normal`, but I get the error, `error: file '/boot/grub2/i386-pc/normal.mod' not found.`. Running `ls (hd1)/` shows me nothing! Does that mean there's nothing on the drive I need, or just that the information isn't quite accessible? Hmm. I guess a restart couldn't hurt more; let's try that now that the variables are set.

RESOLUTION

Actually, you know what, I think I discovered the mistake that caused this whole mess. I decided to check the box to include the hot spare in the installation process. It has no place here; it's managed strictly by the RAID card. It got assigned `sda` and messed everything up. I'm just gonna turn all this off and redo the installation correctly this time.

6.2 Remote Rocks Servers Inaccessible

Normally, the Rocks rolls are able to be accessed from the remote Rocks server in the "ROCKS ROLLS" section of the installation. After the kerfuffle with the boot loader, however, the server seems to be inaccessible even though the CE has internet, verified by pinging through the console.

I'm going to use the ability of the installer to remotely obtain the rolls to see if the issue is just with the communicating with the server, or if it's a larger problem. I've placed all the required rolls on NAS-1, and I'm going to try accessing them from there.

To do that, the rolls need to be hosted on an HTTP server on NAS-1. I'm installing and setting up Apache.

When I tried to `yum install httpd`, none of the mirrors worked, so the installation failed. I tried a `yum update` and a `yum clean all`, but the clean just confirmed that there was a serious problem. Now yum reports that `mirrors.centos.org` cannot be resolved. `yum repolist all` also runs into the same issue; only the 5 recommended steps are listed.

Turns out `/etc/resolv.conf` was blank, and, amazingly, adding `nameserver 9.9.9.9` to it fixed the problem. I wonder if that same file is messed up in the installer.

On the installer, by changing the kernel parameters from "quiet" to "verbose", or even "debug", the booting up of the GUI installer can be seen and interrupted, providing access to the shell. `ifconfig` shows the four network ports and the local network, just as it should, but nothing is connected to the internet. The cables are plugged in, so let's see about enabling the internet.

The Arch Linux wiki page for network configuration is very detailed and it seems to be helpful. I'm going through it.

I've gone through the page, setting the ip manually and editing `/etc/hosts`, but no luck. I'm going to try the `/etc/resolv.conf` thing that worked on NAS-1.

The Rocks installation CD on the CE suddenly has internet! The IP is 163.118.42.1/25, the broadcast IP is 163.118.42.127, and the gateway IP is 163.118.42.126

Enable Internet in Boot CD

```
ip address add 163.118.42.1/25 broadcast + dev enp10s0f0

ip route add default via 163.118.42.126 dev enp10s0f0

echo "nameserver 9.9.9.9" >> /etc/resolv.conf
```

Now the manual installation via anaconda's CLI can begin. It is difficult to test individual lines of code and functions because of so many references to "self" in the code, which indicates an incredible reliance on being in a special running environment to operate properly.

I'm trying something new. I'm installing Rocks onto a flash drive on another machine, then I'm going to `dd` the flash drive image onto the hard drive in the cluster. Maybe that will work.

Completely Wipe USB

```
parted /dev/sd<x> mklabel loop
```

To get to the anaconda command line application:

- load the Anaconda GUI like normal
- Ctl-Alt-F2 to drop to the base terminal of the CD
- try to run `anaconda`
- when X fails to load properly, select the VNC option
- Although Anaconda will continue to not start up properly, the Anaconda environment (with the tmux bar at the bottom) will load.

Just to try something new, I loaded a ubuntu liveCD onto the CE and had a poke around. It looks like everything is where it used to be on the partitions of the original drive, `/dev/sdb2`, specifically the `/root` partition. A curious note about how those partitions are organized: `/dev/sdb2` is

formated as a Linux logical volume, and the logical partitions of that logical volume are found under `/dev/rocks_uscms1/`. `/dev/rocks_uscms1/root` is the original root file system, and I mounted it with `mount /dev/rocks_uscms1/root /mnt/sdb_root`. Something interesting I found, though, is the contents of `/mnt/sdb_root/etc/redhat-release`, which report that CentOS 7 is installed on the system. Hmm. Did the original install do something? Additionally, the `/mnt/sdb_root/boot` directory is blank. While THAT boot directory is blank, however, the specific boot partition of `/dev/sdb`, `/dev/sdb1`, is fully populated with an `el7` image and an updated grub.

To enable ssh on a Ubuntu LiveCD, install `openssh-server` with `sudo apt install openssh-server`, and check to make sure the service is running with `sudo service ssh status`.

BREAK-THROUGH! After fiddling with GRUB from the LiveCD, namely reinstalling it onto the boot partition, the CE still refused to boot! This is because the 3Ware card was not configured to be booted into on the boot order section of the BIOS. This page is accessed by slamming the DELETE key immediately upon turning on the CE. The 3Ware card had its boot status elevated, and I was presented with a fully fleshed out GRUB prompt! (In contrast to the extremely limited rescue GRUB prompt from earlier.) What is available is summarized in Table 6.2.

Table 6: A list of the partitions available to us at the GRUB prompt.

Partition	Filesystem	Summary of Contents
`(proc)`	`procfs`	none
`(hd0)`	n/a	n/a
`(hd0,msdos2)`	n/a	n/a
`(hd0,msdos1)`	`ext*`	boot partition of CE data drive
`(hd1)`	`ext*`	none
`(hd1,msdos1)`	`ext*`	misc. config files (namely Squid)
`(fd0)`	n/a	n/a

Let's try to boot into the boot partition. Following some instructions from online, the first step is to load the `ext2` module, `insmod ext2`. Next, the root of the prompt needs to be set to the boot partition, `set root='(hd0,msdos1)'`. If the root was set correctly, the output of the `ls /` command ought to be identical to the output of `ls (hd0,msdos1)`. In the new `/`, there should be two files that start with `vmlinuz`. We're only concerned with the one that does not have the word "rescue" in the name. To load the compiled kernel, `linux <non-rescuse vmlinuz>`, then to boot it, `boot`.

Unfortunately, the kernel panics when it tries to boot; the rescue kernel was also tried to no avail. The rescue kernel has the panic `not syncing: VFS: Unable to mount root fs on unknown-block(0,0)`. A quick search has revealed that it might be complaining about not having the `initramfs` loaded beforehand. We have that file, but how do we load it properly?

I wonder what happens if we use one of the `initramfs` files with `initrd` instead of just the plain `initrd` image. Also, I'm gonna try to boot the rescue pair. It worked! I've booted into the rescue image! I think that might mean the images aren't ENTIRELY busted.

Something interesting I've noticed: the `/sysroot` directory seems to be the miscellaneous configuration files from `(hd1,msdos1)` from the GRUB. We've also got a `/dev` directory with two drives, one with one partition (`sda`) and another with two (`sdb`). `mount` reveals that `/dev/sda1` is mounted on `sysroot`, and that its filesystem is `ext3`. `/dev/sdb` isn't mounted at all. I've mounted `/dev/sdb1` on a directory I created, `/mnt/sdb1`, and it's the boot partition of the data drive, `(hd0,msdos1)` from the GRUB. I tried to mount `/dev/sdb2`, but it complained that its filesystem was of type `LVM2b3_member`, which means that it's the root filesystem of the CE. Unlike the Ubuntu LiveCD, however, there is no

`/dev/rocks` directory where those logical partitions would be stored. That's enough poking around for now, let's try to load the proper initramfs-kernel pair.

What I thought was rescue mode is actually emergency mode; the regular images brought me to the same place. What can I do with emergency mode?

The issue with the logical partitions may be a GRUB configuration issue. What if GRUB just needs to be configured to look into and properly load the logical partitions? When the physical boot partition is loaded, the complaint is that the `/etc/os-release` file is not found. That's fine because that file is found on the root partition. In our case, however, that partition is logical. How can we configure GRUB to boot into a logical partition?

RESOLUTION

The CE was not getting internet because a DNS server was not specified in the "Network Configuration" section in the Anaconda installer. Throwing `8.8.8.8` in the DNS slot fixed the problem! Internet is back!

6.3 Nodes not found in `insert-ethers`

I found the documentation from when Ankit and Christian installed Rocks 6 on all the nodes back in 2014. They said, after running `insert-ethers --cabinet=1` on the CE and selecting *Nodes*, they inserted the Rocks 6 disc into a node and booted into it. When the disc was loaded, the node sent out a DHCP request that was then picked up by insert-ethers. Let's see about trying that out on compute-1-0. Since we don't have a Jumbo DVD for Rocks 7.0, I'm trying with the Rocks 7.0 kernel disc.

The kernel disc booted to the language selection screen as expected. Unfortunately, however, I seem to be stuck here. The screen requires that a mouse be used to select the *Continue* button, but the mouse doesn't work (USB or PS2). *sigh* I'm gonna see about restarting the node and interrupting the GUI installer like I did before so I can attempt a command line installation.

To do a command line install, the GUI installer must be interrupted before it can start up. When greeted with the first splash screen, press TAB to edit the kernel settings: change "quiet" to "debug" so that more text will appear on the screen during boot which will allow for more time to cancel the installer startup with control-C. With the GUI installer interrupted, navigate to the shell and run `anaconda -T` to begin the text-based version of anaconda.

Be sure to choose an appropriate time zone. Change the *Software selection* from "Minimal Install" to "Compute Node". Select *Network configuration*, then the correct ethernet device (most likely `enp4s0f0`). Configure the device to automatically connect after reboot and to apply configuration in the installer. The defaults for the other settings ought to be sufficient. Ensure that the software selection was successful, create a root password, then begin the installation.

The installation failed; it's complaining about an attribute not having the expected data, "AttributeError: 'RocksRollsData' object has no attribute 'info"'. Checking back on `insert-ethers`, however, something interesting has revealed itself! `compute-1-0`, the node on which the installation failed, is now visible. The "()" is empty, which indicates that, while the node is visible, it has not requested a kickstart file. I'm guessing, if it worked properly, the "*" would appear between the "()" on its own. Since that hasn't happened here, I'm assuming it didn't work, which would make sense since the installation failed. Let's try again to see what happens.

Alright, new problem: `insert-ethers` won't close and `tmux` window switching won't work. I guess I ought to restart the system (Ctrl-Alt-Delete will bring up a GUI shutdown/restart menu).

After a restart, `insert-ethers` crashes on startup. It reports "Access denied for user 'root'@'localhost"', which is more than a little concerning. What happens if I try a hard restart? Let's find out. No change. Huh, what an interesting problem to have.

Sam did some MySQL nonsense (detailed in the adminlog) to get a new error: "error - unable to download kickstart.". It also suggests to verify that `httpd` is running; it, in fact, is. She also found a website that recommends a course of action in response to this error. Unfortunately, turns out the site is 6 years old, and not of much help.

Some disappointing news: I read through Ankit's old documentation from when he and Christian were building the cluster with Rocks 6 back in 2014. They were trying to set up the wiki and the nodes at the same time, and the MySQL didn't like that so much. Sounds familiar! They had to reinstall the CE, so that's what we're gonna go do.

I've backed up the adminlog, thrown in the Rocks 7 kernel CD, and restarted the machine. I'm going to follow the installation instructions at the beginning of this document.

We've thrown the same Rocks 7 on the node that's on the CE. We were hoping to get a Rocks software package "compute node" by pointing the installation source at a specific URL, but the only available one is "minimal install".

We're gonna try installing the node from itself.

As a last ditch effort, I'm going to try to follow the directions exactly on a brand new node. We started `insert-ethers`, selected "Compute", and booted the new node, `compute-1-2` (the real one), into the Kernel Roll CD. We have arrived at the initial Rocks 7.0 screen where we choose to Install Rocks or test the media. We've selected the normal "Install Rocks" option. Upon turning on the network under the "Network Configuration" section of the installer, `insert-ethers` reports having received a DHCP request from `compute-0-1`, what `compute-1-2` has decided to call itself (this is confirmed by what the installer reports to be the "Current host name". We noticed something very suspicious on the `insert-ethers` page; it said "Opened kickstart access to `10.1.1.0/255.255.255.0` network". Hmm, what if we point the "Installation Source" of the installer at that address? After trying the URLs `10.1.1.0/255.255.255.0` and `10.1.1.0`, we just threw in `255.255.255.0` and it seemed to sit longer than the other ones. While it was doing its thing, I noticed the "Cluster Private Network" section had yet to be configured. I clicked that section and, after verifying that the default values were in fact the proper ones, clicked "Done", which saved the configuration. After that had been completed, the grayed-out progress text for the "Installation Source" changed from "Probing storage..." to "Setting up installation source". It's been sitting at there in that state for quite some time. A'ight, it's been sitting here for far too long; we're gonna try the same thing on the real `compute-1-3`. This time, we'll configure the private network before messing with the "Installation Source".

Alright, we've started up the new node, and its calling itself `compute-0-2`. `insert-ethers` has picked it up. We've selected the "Cluster Network" section, verified that the following settings are acceptable:

Table 7: The default "Cluster Network" settings.

Private Cluster Interface	`enp4s0f1;00:30:48:C2:F4:41`
Private Domain Name	local
MTU	1500
IPv4 Address	10.1.1.1
IPv4 Netmask	255.255.255.0

Selecting "Done" saves these settings and completes the configuration. Now let's do the "Installation Source". We're gonna point the node at the full address given to us by `insert-ethers` using HTTP: `10.1.1.0/255.255.255.0` (the use of HTTP was garnered from miscellaneous brief mentions in online forums). When "Done" is selected, the "Installation Source" attempt to probe the storage, but it

ultimately fails. Let's try just `10.1.1.0`. That failed too, now just `255.255.255.0`. I discovered something interesting: running `rocks list network` on the CE gives the same network provided by `insert-ethers`. What if the "Cluster Private Network" section needs to be configured so that it matches THAT network? Let's try that on the real `compute-1-4`.

This one's calling itself `compute-0-3`. I've changed the "IPv4 Address" in the "Cluster Private Network" section from `10.1.1.1` to `10.1.1.0`. No dice; fails when pointed at `10.1.1.0/255.255.255.0` and `10.1.1.0`. Actually, rereading the documentation, it says that kickstart files are transferred over HTTPS rather than HTTP. Let's try that setting. Nope, nothing. I'm gonna send the

We've made a Hypernews post asking for help. In the mean time, we're putting Rocks Kernel on a node again to play with for a bit. When we turned it on, it complained of a kernel error and said to run `abrt-cli list` to view it. The reason it gave was "nobody cared (try booting with the `irqpoll` option". A quick search revealed that the error means an interrupt was not handled, usually a symptom of buggy firmware. To boot with the `irqpoll` option, which will, when an interrupt is encountered, poll all interrupt handlers in an attempt to resolve the interruption.

While it doesn't seem like the CE and the Rocks node can see each other (they have both assigned themselves a local IP of 10.1.1.1), they can both see a machine with local IP 10.1.1.254 (MAC: 00:25:90:33:A3:D8). What is this machine? It's not NAS-0, the SE (which is off), or the network switch. Also, what's `compute-0-0`? Well, mystery solved; I ssh-ed into 10.1.1.254 to find that it is NAS-1.

We're gonna try wiping all the stuff `insert-ethers` has done and try again. The command is `insert-ethers --remove <host name>`, and the list of host names can be found with `rocks list host`. Now that `insert-ethers` has been cleared, let's see what we can do with it now. It was also recommended that `httpd`, `dhcpd`, and some other services be synced and restarted:

- `service dhcpd restart`
- `service httpd restart`
- `service foundation-mysql restart`
- `service autofs restart`
- `rocks sync config`
- `rocks sync users`

Just to see what would happen, I then ran `insert-ethers` on the CE and restarted `compute-1-0`, the node with the Rocks Kernel on it. Nothing happened with that, so I booted it back into the Rocks Kernel CD. This time, when we turned networking on, it called itself `compute-0-1`! That's still not `compute-1-0`, but it's closer. Also, `insert-ethers` has seen `compute-0-1`. In the Anaconda installer we manually changed the host name from `compute-0-1` to `compute-1-0`, but that change was not reflected in `insert-ethers`; we're not even sure if that change was real.

RESOLUTION

Daniel Campos is investigating the cluster. He's gotten into the BIOS of `compute-1-0` by spamming DELETE. The nodes DO have support for PXE boot! He changed the boot order to prioritize the ethernet port. On the head node, he set up `tcpdump` to monitor network traffic to see if the head node will pick up anything from the node. It booted into the Anaconda installer after downloading the kickstart file from the head node, and it said it got all the configurations from the kickstart file!

6.4 NAS-0 RAID Issues

After discovering a drive failure in NAS-0, we tried to replace it with a new drive and were surprised to find that it could not discover the new drive. Upon further examination, we found that the newly inserted drive decided to become a part of unit 1 rather than rebuild itself as part of unit 0, the mirror array. Now the RAID card thinks it has 1 degraded mirror array and 15 JBOD units; that's a total of 17 drives, even though there are only 16 drives plugged in. NAS-0 can't do math! Hmm. Let's see what ZFS says about all this.

On startup, we got a ZFS error: the zpool `nas0` couldn't be imported because `/var/lib/dkms/zfs/0.7.5/source/dkms.conf` doesn't exist. To see if a `dkms.conf` exists anywhere on NAS-0, we ran:

```
locate dkms.conf
```

```
/usr/src/spl-0.7.5/dkms.conf
/usr/src/spl-0.7.9/dkms.conf
/usr/src/zfs-0.7.9/dkms.conf
/var/lib/dkms/zfs/0.7.9/build/dkms.conf
```

What's suspicious here, is that several of these files seem to be of a newer version, 0.7.9 rather than the apparently expected 0.7.5.

To investigate `dkms` itself, we ran `dkms status`, which, again, reported that the `dkms.conf` file didn't exist, but it also warned that the built and installed modules are different! I suspect a version conflict because `dkms status` identifies as version 0.7.5, while all but one of the `dkms.conf` files available are version 0.7.9. Additionally, the `/var dkms.conf` file is under a `build` directory, which gives further credence to `dkms status`'s warning.

We embarked upon a spelunking expedition to investigate what happened to `/var/lib/dkms/zfs/0.7.5/source/dkms.conf`, and found something of interest!
Within `/var/lib/dkms/zfs/0.7.5`, the `source` "directory" is actually a broken symbolic link pointing to `/usr/src/zfs-0.7.5`, where the other `dkms.conf`s are found! Unfortunately, however, there is only a `/usr/src/zfs-0.7.9` directory, while `/usr/src/zfs-0.7.5` is suspiciously absent.

ASIDE

Before proceeding, we performed a quick check to make sure NAS-0's data was all nicely backed up on NAS-1. Always make sure the data's backed up before messing with where it's stored!

We have a question to answer to help us diagnose the cause of the problem: Is ZFS telling DKMS to look in the wrong spot, or is DKMS looking in the wrong spot all on its own? Let's find where DKMS is told to check for `dkms.conf` at `/var/lib/dkms/zfs/0.7.5/source/dkms.conf`.

Just to see what would happen, we changed the broken symlink in `/var/lib/dkms/zfs/0.7.5` from `source -> /usr/src/zfs-0.7.5` to `source -> /usr/src/zfs-0.7.9`. `dkms status` now successfully adds the ZFS module, despite continued warnings proclaiming that the built and installed modules are different. Although the ZFS module has been added and the commands work, `zpool list` returns no pools! Huh. Does this have to do with our jank method of making DKMS work, or is there something more sinister going on?

Now that `dkms.conf` can be found, let's restart NAS-0 to see what the ZFS boot message says now. The ZFS error saying it couldn't import the pool `nas0` is still there, but DKMS now runs some pre-build script that appears to hang at `checking spl build directory...`. After sitting there for a while, it reported that the build directory was `/var/lib/dkms/spl/0.7.9/2.6.32-431.11.2.el6.x86_64/x86_64`,

then threw a configuration error saying to make sure that the `kmod spl devel <kernel>` package is installed for our distribution; it failed to find `spl_config.h` in either
`/usr/src/spl-0.7.9/2.6.32-431.11.2.el6.x86_64` or `/usr/src/spl-0.7.9`. It then failed to build the ZFS module and continued the normal booting process.

`dkms status` reports that SPL 0.7.5 and ZFS 0.7.9 have been added, but it warns that there is a difference between the build and installed modules of SPL.

While `zpool status` shows that there are no pools available, `zpool import` happily shows us that the `nas0` pool is alive and well in the ONLINE state. The "action" says that "The pool can be imported using its name or numeric identifier.", which implies that the pool need just be imported to work. `zpool import nas0` seems to have done the trick; `zpool status` shows the status of the data drives. While all the main drives are reported to be online, the two hot spares, `sdo` and `sdi`, have failed; they're state is FAULTED. Another peculiar note is that, while most of the drives are represented by their SCSI identifiers, two are simply identified by their names, `sdg` and `sdh`. The drives both have the status ONLINE, however, so this may not be an issue. Interestingly enough, `tw_cli /c0 show` does not show that the two hot spares are damaged; they appear fine to the RAID card. Perhaps the two drives are merely suffering from a ZFS issue that can be easily corrected? Additionally, `tw_cli /c0 show` reports a further drive failure in `p5`. Unfortunately, it insists on reporting the RAID-1 OS array as degraded while assigning each individual drive its own individual unit.

While investigating `/dev/sd*`, we noticed that `/dev/sdb` and the other drives all had two partitions, `/dev/sdx1` and `/dev/sdx9`, while `/dev/sda`, the drive we presumably just replaced, only has one partition, `/dev/sda1`.

The root issue still appears to be that the new, replacement drive `/dev/sda` in port 0, has been mislabeled as part of unit 1, when it ought to be part of unit 0 alongside the drive in port 1, `/dev/sdb` to form a RAID-1 array. The RAID-1 array still exists as unit 0, albeit in a DEGRADED state for the time being. Let's see what we can do about changing `p0`'s unit affiliation from unit 1 to unit 0.

We found some interesting instructions! It looks like we can delete the extra unit, unit 1, then tell the drive to make itself a part of the degraded array. Let's give it a shot! First, we delete unit 1: `tw_cli maint deleteunit c0 u1`. Then, we explicitly begin the rebuild: `tw_cli /c0/u0 start rebuild disk=0`. Unit 0 is being properly rebuilt! The status of the rebuild process can be monitored with `tw_cli /c0/u0 show rebuildstatus`.

The drive was successfully rebuilt! Unfortunately, two of the drives in the first `raidz2` group decided to fail, `/dev/sdg` and `/dev/sdh`. We threw the two spares in to replace them with `zpool replace nas0 <number of failed drive> sdx`, where `sdx` is the identifier for the spare. While those were resilvering, yet another drive, `/dev/sdf`, failed in the same `raidz2` group as the other two failed drives. We've now got three drives rebuilding in one `raidz2` group, which has a tolerance of two drive failures. The progress of the scan is still increasing, however, so we remain hopeful.